Apple Push Notification Service

Wireless notifications for apps

Session 120

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

Apple Push Notification Service

UIKit API and best practices

Chris Marcellino
iPhone Software Engineer

Introduction

- Make your apps network connected
- Get users the information they want, when they want it
- Minimize impact on battery life

What You'll Learn

- How to push sounds, badges, and alerts
- Server protocols, certificates
- Best practices

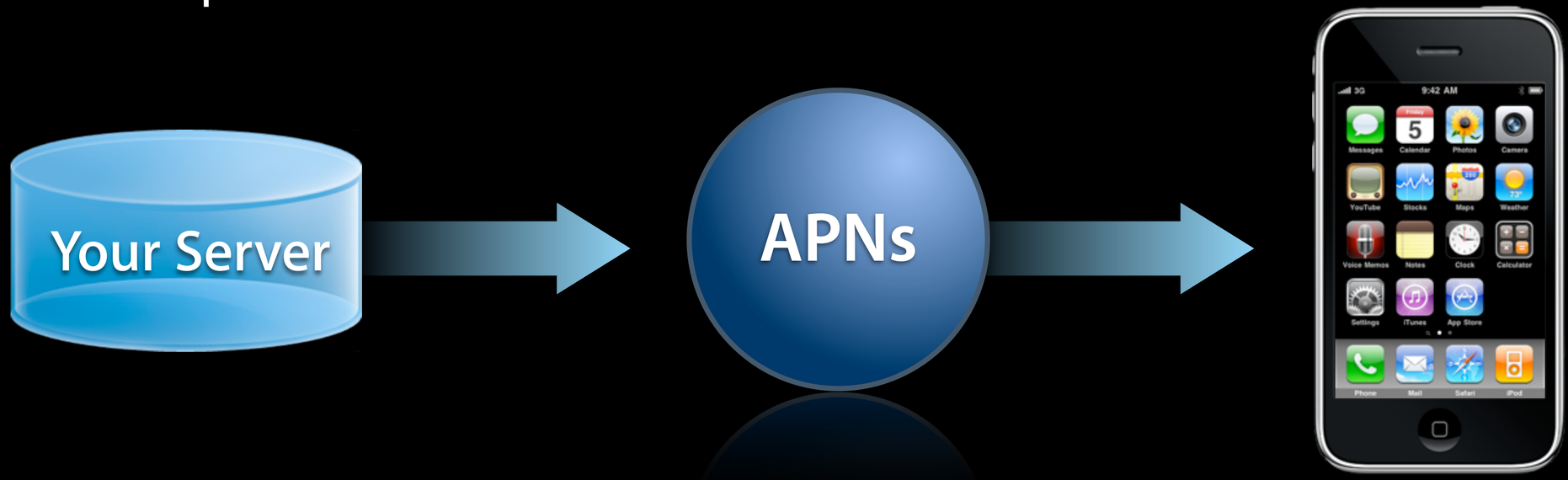
Works over any connection

iPhone Developer Program

It’s Easy

Overview

- One persistent connection per device
- Any number of applications
- Preserves battery life
- Maximizes performance



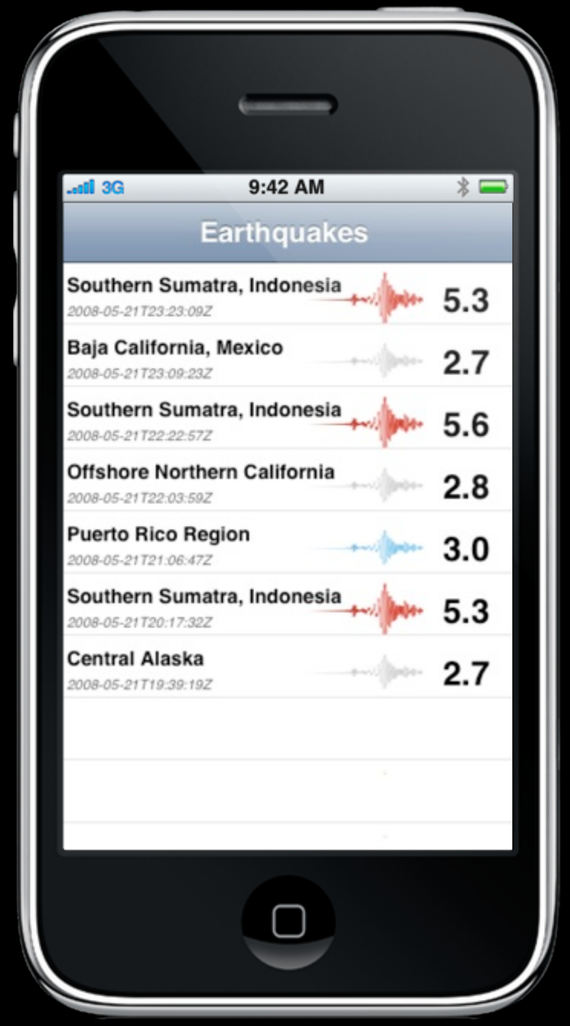
Using the Service

What you need

com.foo.app

Using the Service

What you do



Using the Service

1. Register with the service

Using the Service

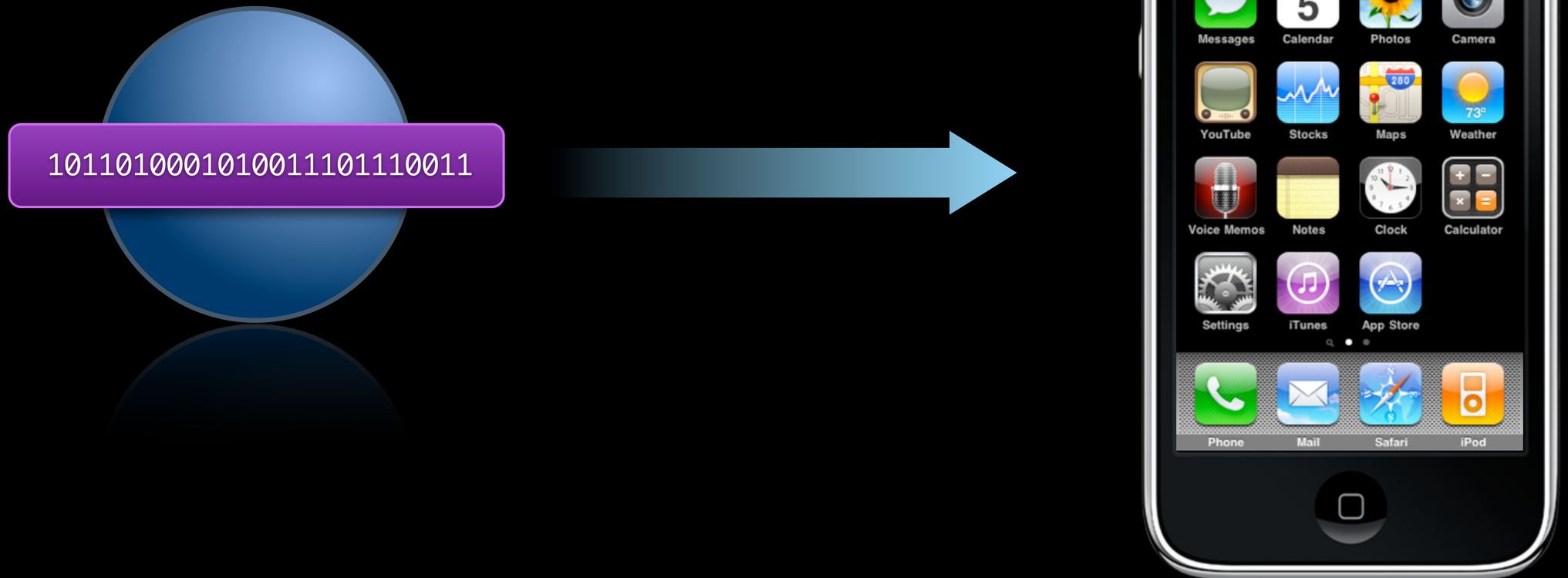
2. Send token to your server

Using the Service

3. Send notifications

Using the Service

4. Receive notifications



Using the Service

1. Register with the service

Registering with the Service

Application launch

- UIKit API in UIApplication.h to register
  - Pass the types you want to receive

```
-(void)application:(UIApplication *)application
     didFinishLaunchingWithOptions:(NSDictionary *)options
{
    // Register this app on this device
    UIRemoteNotificationType myTypes = UIRemoteNotificationTypeSounds |
                                       UIRemoteNotificationTypeBadges;
    [application registerForRemoteNotificationTypes:myTypes];
}
```

Registering with the Service

Delegate callbacks

```
 - (void)application:(UIApplication *)application
    didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)token
{
    // Phone home with device token
}
```

```
 - (void)application:(UIApplication *)application
    didFailToRegisterForRemoteNotificationsWithError:(NSError *)error
{
    // Oh noes! Check your Provisioning Profile and/or entitlements
}
```

Registering with the Service

The device token

- Uniquely identifies device
  - Separate from -[UIDevice deviceIdentifier]
- Call registration API if token is needed
  - Don’t store token on disk

```
96385da767191121a851963983fdac9bbdf74dcf6219ae14ed8d082288aaebaf
```

Registering for Notifications

Optional callbacks and methods

- UIApplicationDelegate

```
- (void)application:(UIApplication *)application
     didReceiveRemoteNotification:(NSDictionary *)userInfo
```

- UIApplication

```
- (UIRemoteNotificationType)enabledRemoteNotificationTypes
```

Using the Service

2. Send token to your server

Using the Service

3. Send notifications

JSON: JavaScript Object Notation

No JavaScript required!

- Human readable
- Compact
- Great for Cocoa property lists

Sending Notifications

Message payload

- Strict RFC 4627 JSON
- 256 byte maximum

```
{
    "aps" : {
        "alert" : "Jen: Sushi at 10?",
        "badge" : 1,
        "sound" : "Jingle.aiff"
    },
    "acme1" : "conversation9964"
}
```

Sending Notifications

Message payload

- *aps* dictionary reserved for the sound, badge, or alert keys
  - All keys optional

```
{
    "aps" : {
        "alert" : "Jen: Sushi at 10?",
        "badge" : 1,
        "sound" : "Jingle.aiff"
    },
    "acme1" : "conversation9964"
}
```

- Rest of payload is for your app

Badges

badge key, integer value

- Positive integer
  - Or omit to remove

```
{
    "aps" : {
        "badge" : 1
    }
}
```

Sounds

sound key, string value

- Either a filename in app bundle
  - linear PCM
  - MA4
  - µLaw
  - aLaw
- Or “default”
- Vibration is automatic

```
{
    "aps" : {
        "sound" : "Jingle.aiff"
    }
}
```

Alerts

alert key, string, or dictionary value

- Simplest form is just a string value

```
{
    "aps" : {
        "alert" : "Jen: Sushi at 10?"
    }
}
```

Alerts

Localized alert body and button titles

- Replace the *alert* string with a dictionary

```
{
    "aps" : {
        "alert" : {
            "loc-key" : "MSG_POSTED",
            "loc-args" : [ "Jen", "iPhone" ],
        },
    }
}
```

English.lproj/Localizable.strings

```
“MSG_POSTED” = “%@ has posted a message for you about %@”
```

Spanish.lproj/Localizable.strings

```
“MSG_POSTED” = “%@ acaba de fijar un mensaje para usted sobre %@”
```

Alerts

Localized alert body and button titles

- Same syntax as NSLocalizedString() and -[NSString stringWithFormat:]
  - Arguments can be reordered by localizers

English.lproj/Localizable.strings

```
“MSG_POSTED” = “%@ sent you a message about %@”
```

Latin.lproj/Localizable.strings

```
“MSG_POSTED” = “Lorem ipsum dolor %2$@ sit amet adipiscing elit %1$@.”
```

- Keep the localized string keys short!

Alerts

Customize the view button by using *action-loc-key*

```
{
    "aps" : {
        "alert" : {
            "body" : "John has invited you to chess",
            "action-loc-key" : "PLAY_KEY"
        }
    }
}
```

```
{
    "aps" : {
        "alert" : {
            "body" : "Your iPhone loves you!",
            "action-loc-key" : null
        }
    }
}
```

Sending the Payload

Send JSON that is stripped of whitespace

```
{"aps":{"alert":"Jen: Sushi at 10?","badge":1,
"sound":"Jingle.aiff"},"acme1":"conversation9964"
}
```

```
{
    "aps" : {
        "alert" : "Jen: Sushi at 10?",
        "badge" : 1,
        "sound" : "Jingle1.aiff"
    },
    "acme1" : "conversation9964"
}
```

196 bytes

Apple Push Notification Service

Server connectivity and best practices

Kris Macleod Bell
Server Engineering Lead

Agenda

Sending messages

- Prerequisites
- Service overview
- Delivery steps
- Connections
- Redelivery
- Feedback
- Scaling
- Best practices

Recap

Registration

Token

Your Content Server

Sending Messages

Prerequisites

X.509 SSL certificate

Message payload

```
{
  "aps":  {
    "alert": "I CAN HAZ PUSH?",
    "badge": 1,
    "sound": "meow.aif"
  }
}
```

SSL Certificate

Your access to the service

- Your *identity*
  - Identifies your application
- Establishes trust
  - Keep it safe!
- Certificate revocation
  - See ToS

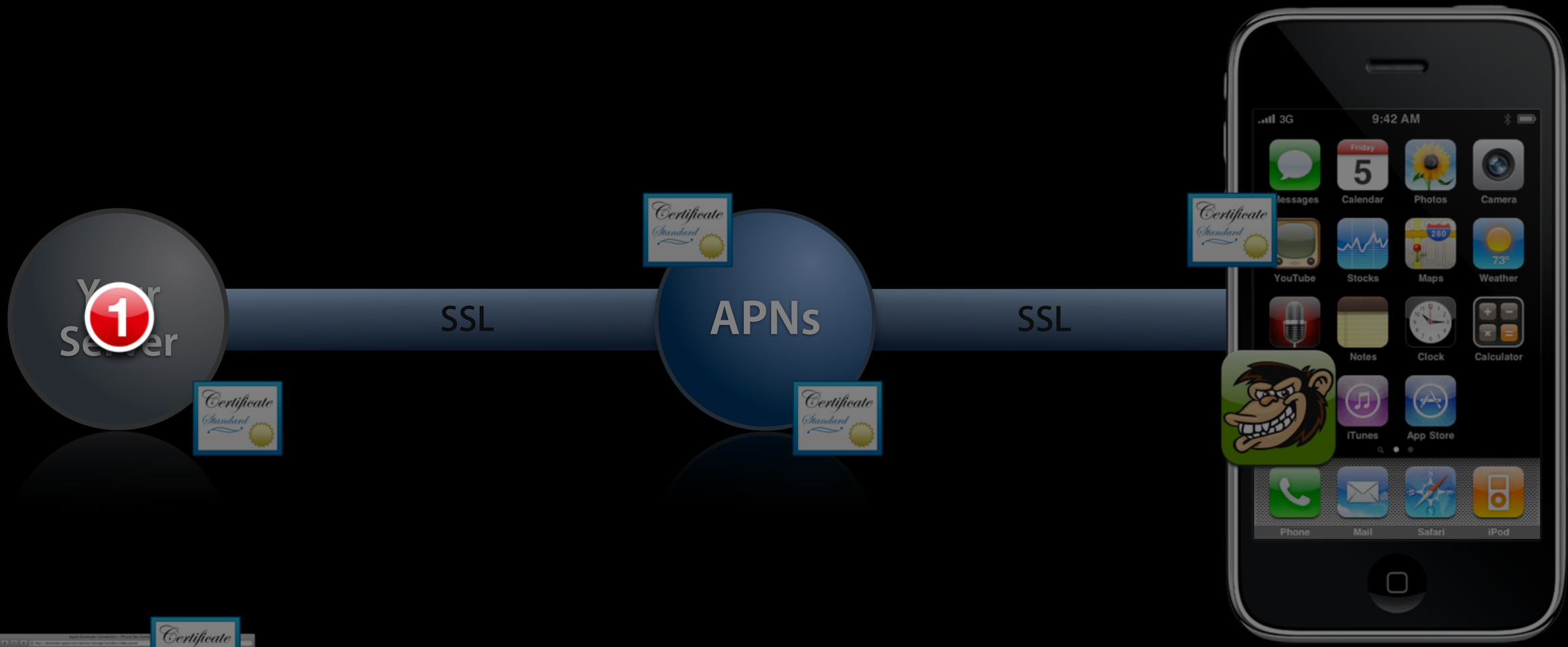
Message Content

RFC 4627 JSON

- 256 bytes max
- Avoid sensitive data

```
{
  "aps":  {
    "alert": "Purchase approved",
    "badge": 1,
    "sound": "kaa-ching.aif"
  },
  "CC":  "5555-4444-3333-2222"
}
```

Service Overview



Bundle ID

Sending Messages

Delivery steps

- Open SSL connection (code on dev forums)
  - Multiple connections
  - Keep connections open
- Construct message
  - JSON
- Send notification

Sending Messages

Persistent connections

- Keep connections open
  - Do *not* open/close per notification
  - Don’t be mistaken for DoS
  - Severed connections?
- Exceptions
  - Inactivity timeout
  - Reopen as necessary

Sending Messages

Message transfer

- Binary protocol
  - High throughput
  - Low overhead
  - Streaming
- Simple format…

Sending Messages

Binary format ~ see “Programming Guide”

token length *payload length*

cmd *device token* *payload*

0	0	20	dcaf30b...	0	13	{“aps”:{“badge”:1}}

bytes: *1* *2* *32* *2* *19*

(big endian) *(big endian)*

Sending Messages

Redelivery

- Offline devices
  - One message per application
  - Old messages are discarded
- “Best attempt” delivery
  - Similar to SMS

Uninstalled Applications

Feedback service

- Populates on failed send
  - Clears on fetch
- Regular SSL connection
  - Using same certificate
  - Different host name
- Connection initiates reply
  - No command required
  - Disconnected upon completion

Uninstalled Applications

Feedback format ~ see “Programming Guide”

unix timestamp				*token length*		*device token*
3	fa	7c	a9	0	20	dcaf30b353a009000...
bytes:	*4*			*2*		*32*
	(big endian)			*(big endian)*		

Scalability

Demo

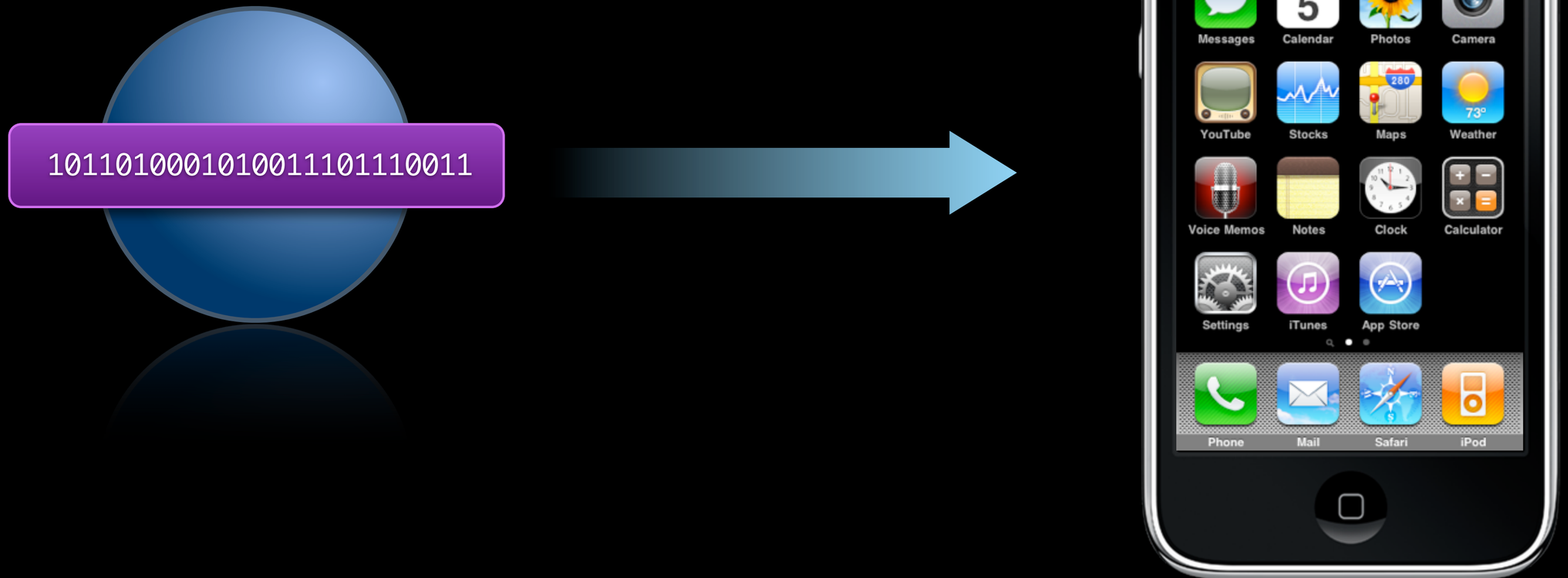
Best Practices

Four things to remember

- Persistent connections
  - Keep them open
- Uninstalled applications
  - Use feedback service
- Avoid sensitive data
  - Notification only
- Protect your identity
  - Represents trust

Using the Service

4. Receive notifications



Receiving Notifications

- iPhone OS does *nothing* if your app is running, except call

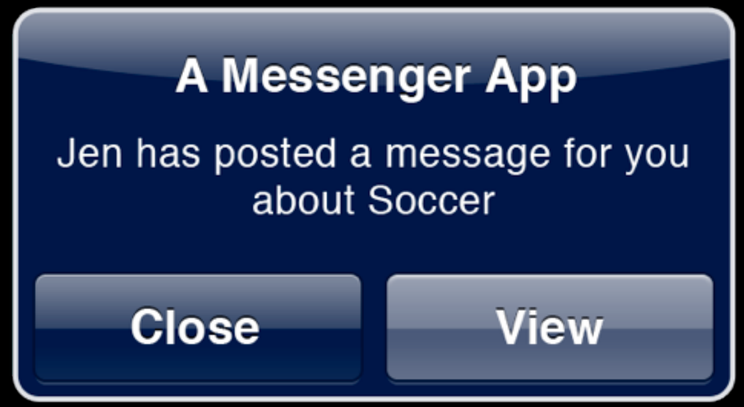
```
- (void)application:(UIApplication *)application
     didReceiveRemoteNotification:(NSDictionary *)userInfo
```

- Up to you to update your UI and play sounds

Getting the Payload

- If your app is running, you'll only get
  - -application:didReceiveRemoteNotification:

- If not, tapping View will launch your app
  - -application:didFinishLaunchingWithOptions:
  - else -application:didReceiveRemoteNotification:



Getting the Payload

Payloads must be stateful

- Queue depth of 1
- Notifications must have complete state
  - Any notification received clears all badges and alerts

Centralized Settings

In the Settings application

- Master switch
- Per-app settings for alerts, sounds, badges
  - Only reflects what you enabled at registration

User Experience

Badges, sounds, and alerts

- Don’t annoy your users!
  - Use alerts sparingly, use badges instead

Badges

We do need them

- Primary notification type
- For actionable information
- Keep a server-side count
- Test your logic
  - Count can change while inactive
  - Notifications can arrive while active
  - Ensure accurate information

Sounds

Context without looking

- Distinguish your app clearly
- Keep sounds short but distinct
  - Branded sounds good
  - Ringtones and songs bad
- Ensure high quality
  - 16-bit, 44.1kHz, stereo AIFF

Alerts

When to use them

- Worth knowing
- Need to know

Alerts

When not to use them

- Don’t need to know
- Too much information

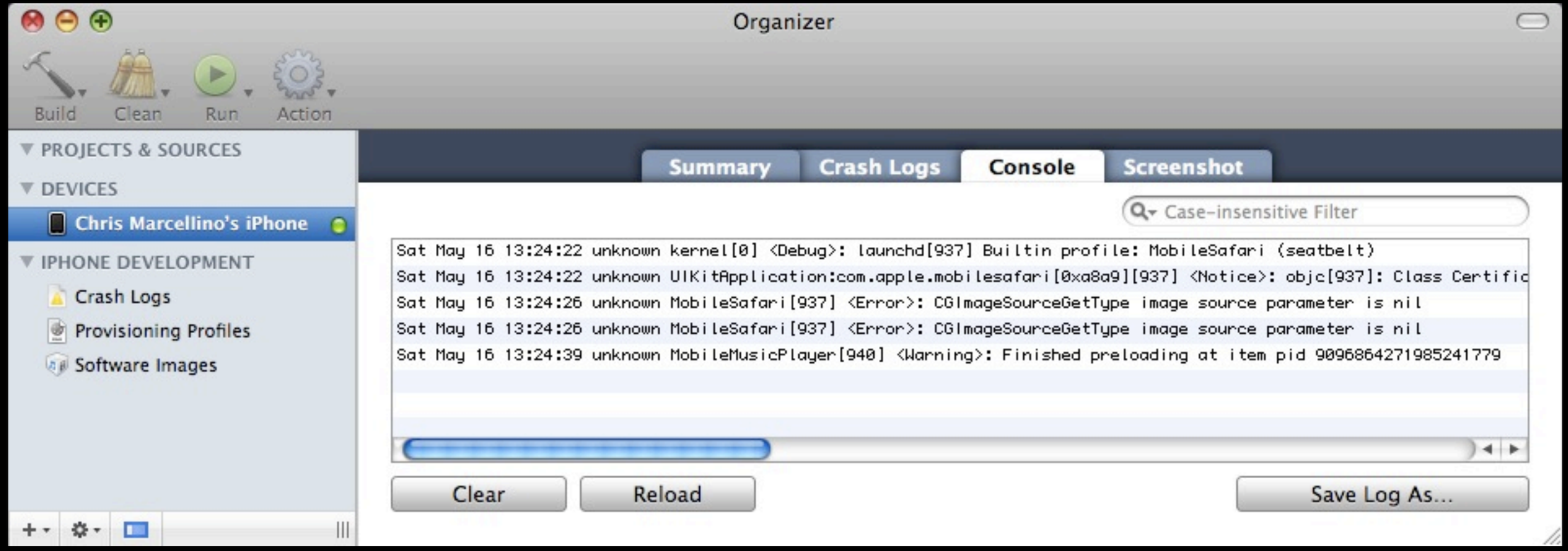
Don’t waste the user’s time

Best Practices

- Pull data while your app is running
  - Since the push service might be turned off
  - Apps should be fully functional without
- Upload the device token to your server often
  - Users can sync and restore your application data without your server's knowledge

Troubleshooting

- Check the iPhone Console
  - In Xcode's Window menu… Organizer… Console tab
  - JSON decoding and sound format errors will be logged here



Troubleshooting

Examples

- What’s wrong here?

```
{
    "aps" : {
        "badge" : 2,
        "sound" : "Jingle.aiff"
    },
}
```

```
apsd[48] <Warning>: Failed to parse JSON message payload for topic 'com.foo.app':
Object value expected for key: aps; Trailing comma disallowed in object
```

Troubleshooting

Examples

- What's wrong here?

```
{
    "aps" : {
        "badge" : 5,
        "sound" : "soundfileA.aiff"
    }
}
```

SoundFileA.aiff

```
SpringBoard[24] <Warning>: No remote notification alert
sound file found for 'soundfileA.aiff'. Playing default sound.
```

Troubleshooting

- Use a library to generate your JSON
- Look at the console for errors
- One more thing…

Demo

Justin Santamaria
iPhone Software Engineer

More Information

Matt Drance
Application Frameworks Evangelist
mdrance@apple.com

Documentation

Apple Push Notification service Programming Guide
http://developer.apple.com/iphone/library

iPhone OS Human Interface Guidelines
http://developer.apple.com/ue

Labs

Apple Push Notification Lab	iPhone Lab C Wednesday 9:00AM
Apple Push Notification Lab	iPhone Lab A Friday 2:00PM

Summary

- Four steps to sending notifications
- Be nice to your users
- Pull data when running
- Keep payloads small
- Keep those server connections up

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09